Architecture des Ordinateurs

Mostafa MRABTI

Plan du cours

1. ***Systèmes de Numération***
2. ***Fonctions logiques élémentaires***
3. ***Fonctions combinatoires usuelles et circuits associés***

Introduction

Signaux analogiques et numériques

Représentation des nombres

- Signal analogique: Grandeur caractéristique du signal porteur de l'information est continue. Infinité de valeurs:

Exemple: Signal sinusoïdal (téléphone fixe)

- Signal numérique: Grandeur caractéristique du signal porteur de l'information est discrète. Nombre fini de valeurs:

Exemple: Signal TELEX

Introduction

Avantages et inconvénients des techniques numériques

- Avantages
  - Simplicité de conception
  - Simplicité de stockage, de reconnaissance et de génération de signaux
  - Plus grande précision.

Introduction

Avantages et inconvénients des techniques numériques

- Inconvénients
  - Complexité des traitements
  - Nécessité de conversion A/N et N/A.
  - Problèmes d'échantillonnage, de quantification: Perte de précision
  - …

Introduction

Le fonctionnement d'un ordinateur est basé sur l'utilisation de la base 2

- Pour cela
  - Représentation des nombres
  - Réalisation des opérations arithmétiques :
    - + et - : facile en entier et en réel
    - x et / : plus difficile
    - Séquenceur, mémorisation…

Introduction

- Tous les ordinateurs sont fondés sur les mêmes principes de base :
  - Programmes / données
  - Processeur / mémoire / périphériques
  - Informations codées en binaire
  - Traitements effectués en binaire

Introduction

- Pourquoi la base 2 ?
  - Simple à implémenter.
- Implémentation technologique :

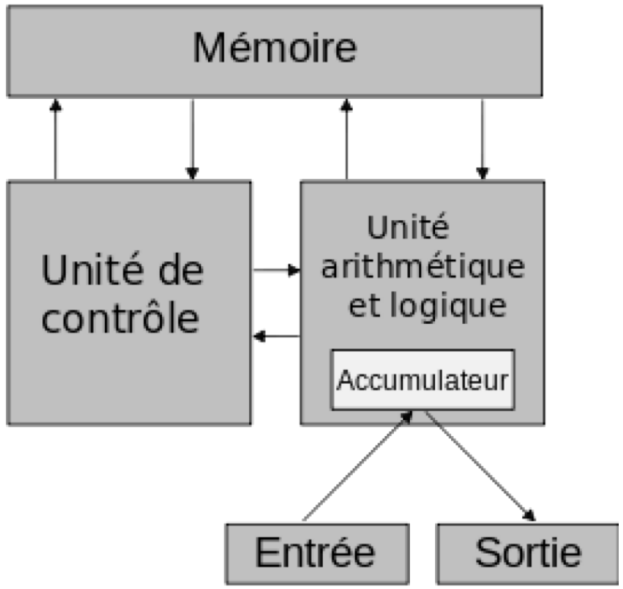
  - transistor - circuit intégré - microprocesseur

  - d ’où : plus rapide, plus petit, moins gourmand, plus fiable, moins coûteux

Introduction

Définition 1

Un processeur est un élément qui exécute des traitements. Les traitements à exécuter sont indiqués par les instructions du programme exécuté. Les traitements s'appliquent sur des données et fournissent également des résultats.

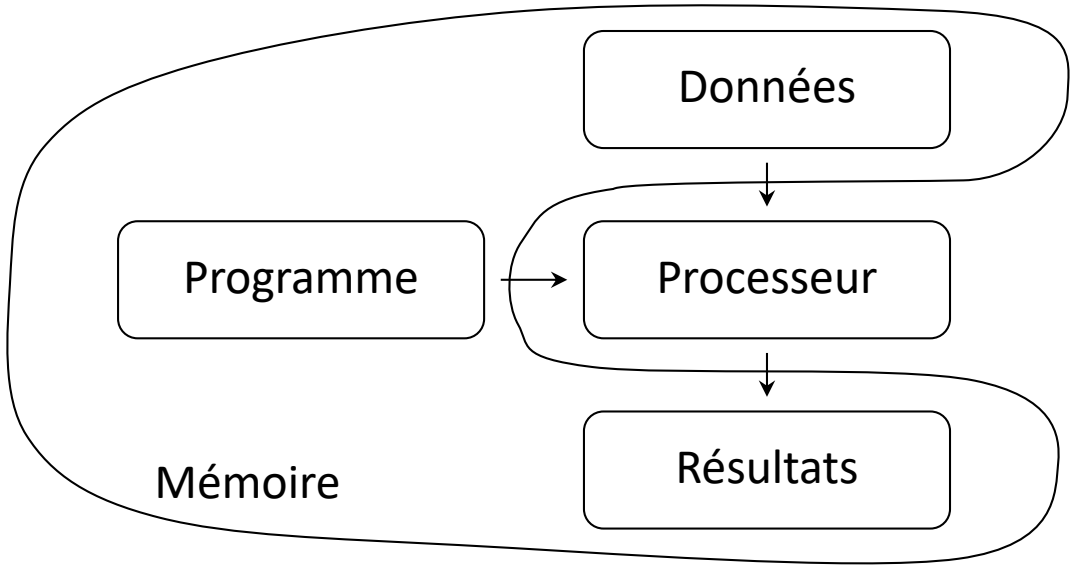


Modèle d'architecture de Von Neumann

Introduction

Définition 2

La mémoire centrale contient les programmes exécutés par le processeur ainsi que les données du programme et ses résultats.



SYSTÈMES DE NUMÉRATION

Systèmes de numération

Définition

- Un système de numération permet de représenter les valeurs numériques le plus simplement et le plus clairement possible.
- Exemples:
  - Système décimal (0,1,2,…9) base 10
  - Système binaire (0,1) base 2
    - Deux états 0 et 1.
    - 0 état bas et 1 état haut: Logique positive
    - 0 état haut et 1 état bas: Logique négative
    - Exemple:
      - » Commutateur ouvert/fermé.
      - » Transistor bloqué/saturé

Systèmes de numération

Différentes bases de numération

- Un même nombre X peut être représenté dans différents systèmes (ou bases) de numération.
- Dans une base B, X s'écrit:
- $X_B = b_{n-1}B^{n-1} + \ldots + b_0B^0 + b_{-1}B^{-1} + \ldots + b_{-m}B^{-m}$

$= b_{n-1} \ldots b_1 b_0 . b_{-1} b_{-2} \ldots b_{-m}$

Partie entière — Pt de séparation — Partie fractionnaire

où $0 \leq b_i \leq B-1$

b_i : Poids de rang i

B: Base

n, m: Nbre de chiffres de X

Systèmes de numération

Différentes bases de numération

- Base décimale: B=10
- $B_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
  - **Exemple:** $1997_{10} = 1.10^3 + 9.10^2 + 9.10^1 + 7.10^0$
  - B=10; n=4; $b_0 = 7$; $b_1 = 9$; $b_2 = 9$; $b_3 = 1$
  - $1997_{10} = b_3 b_2 b_1 b_0$
  - Pour écrire un nombre en base 10, on le décompose en puissances de 10.
  - **Exemple: Nombre décimal**
  - $13.75_{10} = 1.10^1 + 3.10^0 + 7.10^{-1} + 5.10^{-2}$

Systèmes de numération

Différentes bases de numération

- Base binaire: B=2

$B_2 = \{0, 1\}$

Pour écrire un nombre en base 2, on le décompose en puissances de 2.

$19_{10} = 16 + 2 + 1 = 1.2^4 + 0.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 10011)_2$

- Nombre binaire fractionnaire

$1101.101)_2 = 1.2^3 + 1.2^2 + 0.2^1 + 1.2^0 + 1.2^{-1} + 0.2^{-2} + 1.2^{-3}$

$= 8+4+1+0.5+0.125 = 13.625)_{10}$

LSB (Low Significant Byte) : Bit de rang le plus petit

MSB (Most Significant Byte) : Bit de rang le plus grand

Systèmes de numération

Différentes bases de numération

- Base octale: B=8
- $B_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$
  - **Exemple:** $144_8 = 1.8^2 + 4.8^1 + 4.8^0 = 64 + 32 + 4 = 100)_{10}$
- Base duodécimale: B=12
- $B_{12} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B\}$
  - **A=10; B=11.**
- Base héxadécimale: B=16
- $B_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- **A=10; B=11; C=12; D=13; E=14; F=15.**
  - **Exemple:** $1EDF_{16} = 1.16^3 + 14.16^2 + 13.16^1 + 15.16^0$

Systèmes de numération

Passage d'une base à une autre

- Passage de B à B_{10}
  - Permet de déterminer la valeur décimale d'un nombre X écrit dans B quelconque.
- Règle de passage
  - La valeur décimale s'obtient par sa forme polynomiale avec les règles de calcul dans le système décimal.
  - Exemples:
    - $11010011)_2 = 1.2^7 + 1.2^6 + 0.2^5 + 1.2^4 + 0.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 128 + 64 + 16 + 2 + 1 = 211)_{10}$
    - $110011.1001)_2 = 1.2^5 + 1.2^4 + 1.2^1 + 1.2^0 + 1.2^{-1} + 1.2^{-4} = 51{,}5625)_{10}$
    - $4513)_8 = 4.8^3 + 5.8^2 + 1.8^1 + 3.8^0 = 2379)_{10}$
    - $1B20.8)_{16} = 1.16^3 + 11.16^2 + 2.16^1 + 8.16^{-1} = 6944{,}5)_{10}$

Systèmes de numération

Passage d'une base à une autre

- Passage de B_{10} à B quelconque

$$X_B = \sum_{i=0}^{n-1} b_i . B^i$$

- Il faut donc déterminer les poids $b_i \in \{0, 1, 2, \ldots B-1\}$
- Règle 1
  - Dresser une table des puissances de B.
  - À X donné, on retranche la plus grande puissance de B possible.

- ***Exemple:*** conversion de $X_{)10} = 52$ en base 2

$2^0 = 1$; $2^1 = 2$; $2^2 = 4$; $2^3 = 8$; $2^4 = 16$; $2^5 = 32$; $2^6 = 64$ $\quad (2^5 = 32) < 52 < (2^6 = 64)$

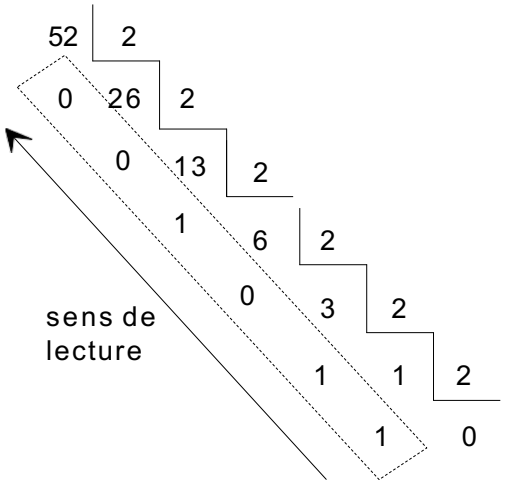
$X - 2^5 = 20$; $\quad 20 - 2^4 = 4$; $\quad 4 - 2^2 = 0$

$X = 1.2^5 + 1.2^4 + 0.2^3 + 1.2^2 + 0.2^1 + 0.2^0 = 110100_{)2}$

Systèmes de numération

Passage d'une base à une autre

- Règle 2
  - On divise le nombre X par B et on conserve le reste (r).
  - On répète le processus en considérant le quotient (q) obtenu.
  - Le nombre X est obtenu en écrivant tous les restes à partir de la fin et de gauche à droite.
  - ***Exemple 1 :*** Conversion de $X_{)10} = 52$ en base 2

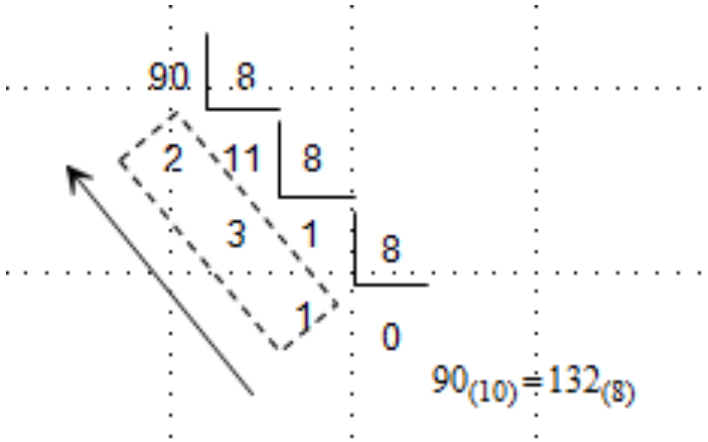


  - $52_{)10} = 110100_{)2}$

Systèmes de numération

Passage d'une base à une autre

- ***Exemple 2*** : Conversion de $X_{(10)} = 90$ en base 8



Systèmes de numération

Système Binaire

Définitions

- Dans le système binaire B=2 ; chaque position binaire (poids) contient un nombre binaire ou bit.
- Bit de l'anglo-saxon : **BI**nary digi**T**
- Le bit est la plus petite quantité d'information dans le système binaire (B=2)

Exemple :

- Bit le plus significatif → 10001011 ← Bit le moins significatif

  MSB: Le bit le + à gauche. LSB: Le bit le + à droite

Décade :	Groupement de 4bits
Octet :	Groupement de 8 bits
Baud (bps) :	nombre de bit d'information transmis ou reçu par seconde
Mot :	Succession de bits traitée comme un tout (mots de 8, 16, 32 bits).

Systèmes de numération

Système Binaire

- **Comptage des chiffres**
- Un chiffre est augmenté de 1 quand tous les chiffres de poids inférieur sont saturés (9 pour B_{10} et 1 pour B_2).

- **Exemple**

B=10	B=2
00	0000
01	0001
02	0010
03	0011
04	0100
05	0101
06	0110
07	0111
08	1000
09	1001
10	1010
etc	etc

Systèmes de numération

Système Binaire

- **Calcul dans le système binaire**
- Le calcul avec les nombres du système binaire est identique à celui avec les nombres d'un systèmes quelconque, mais sont beaucoup plus simples.

Règles de calcul :

Addition

0 + 0 = 0
0 + 1 = 1 + 0 = 1
1 + 1 = 0 + 1 *retenue*
1 + 1 +1 = 1 + 1 *retenue*

Soustraction

0 – 0 = 0
1 – 0 = 1
0 – 1 = 1 + 1 emprunt
1 – 1 = 0
0 – 1 – 1 = 0 + 1 emprunt
1 – 1 – 1 = 1 + 1 emprunt

Multiplication

0 * 0 = 0
1 * 0 = 0
0 * 1 = 0
1 * 1 = 1

Division

Identique à celle du système décimal en utilisant les règles de multiplication et de soustraction déjà définies

Systèmes de numération

Système Binaire

Remarques :

- Ces calculs peuvent être effectués manuellement aussi bien qu'automatiquement grâce à des opérateurs réalisant les quatre opérations sur les nombres.

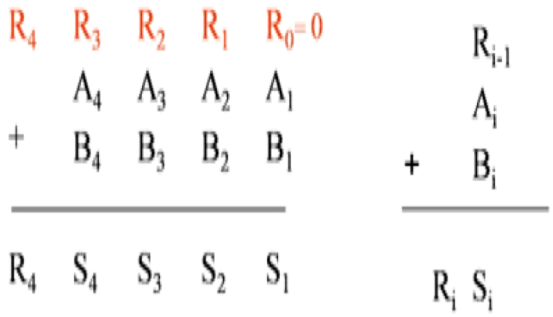
- Dans la suite, seuls les opérateurs d'addition et de soustractions peuvent être traités à cause de la complexité des circuits intervenant dans la multiplication et la division.

Systèmes de numération

Système Binaire

- **Principe d'un additionneur de nombres positifs :**

- Soient deux nombres **A = A_n…A_3 A_2 A_1 A_0** et **B =B_n … B_3 B_2 B_1 B_0**
- La somme **S = A + B** est obtenue par la concaténation des sommes partielles **S_i=A_i+B_i+R_{i-1}** où **r_{i-1}** est une retenue éventuelle de la sommation partielle précédente .

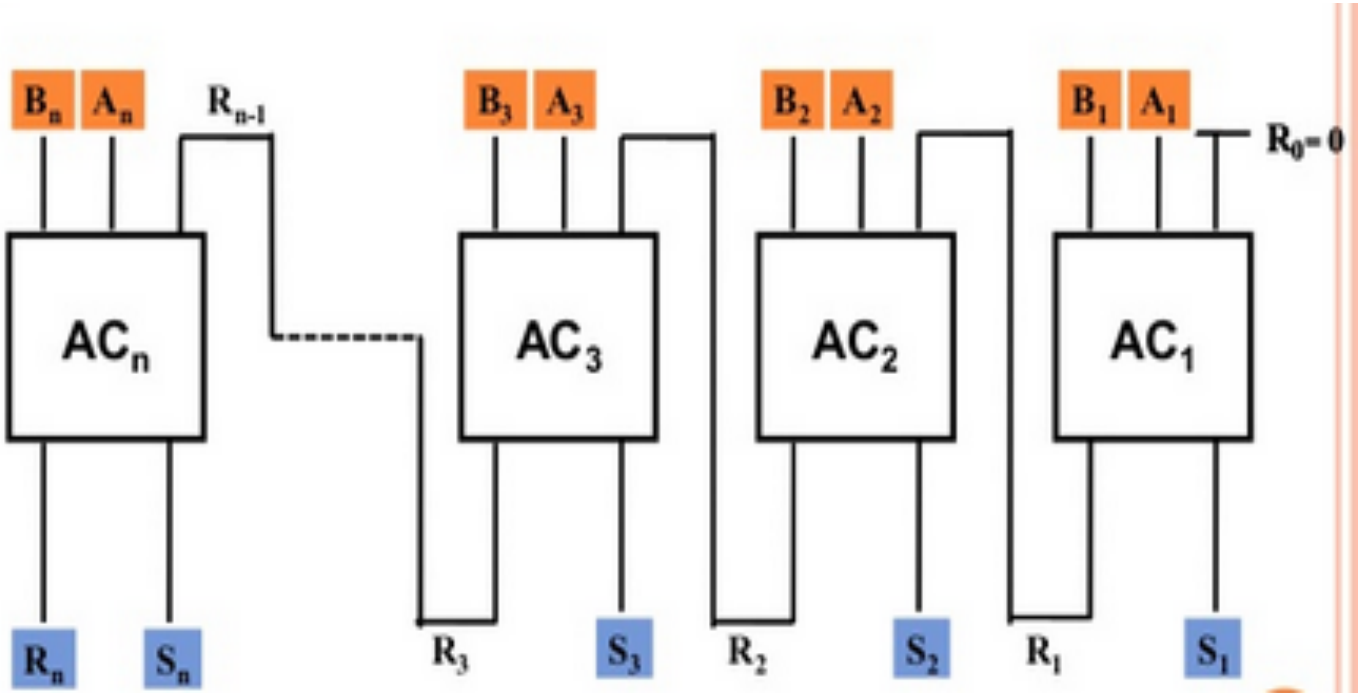


Systèmes de numération

Système Binaire

- **Structure générale d'un additionneur :**

D'après ce qui précède cette structure d'additionneur est :



Chaque bloc a 3 entrées et deux sorties (Si , ri)

Systèmes de numération

Système Binaire

Table de vérité :

4 possibilités pour a_i, b_i

2 possibilités pour r_{i-1}

Entrées			Sorties	
a_i	b_i	r_{i-1}	S_i	r_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

r_{i-1} prend les valeurs 0 et 1pour traiter les 2 cas possibles

appliquer les règles d'addition

Systèmes de numération

Le codage

Codage: Définitions

- Le codage est l'opération de conversion de la forme d'une information.
- Le codage permet de traiter n'importe qu'elle information sous sa forme la plus pratique : Plus simple, plus vite, plus discret, plus économique.
- Les machines numériques ne comprennent que les symboles binaires, d'où le problème de la représentation des symboles et des nombres.

Exemple : Taper sur une calculatrice sin(x)

- Taper sur un clavier $, *, £
- Les codes dont les nombres (digits) peuvent prendre deux valeurs sont dits des codes binaires.
- Un code binaire à n- positions (ou mot de code) permet de représenter 2^n informations différentes.

Systèmes de numération

Le codage

Définitions :

- **Transcodage :** c'est le passage d'un code à un autre
- **Décodage :** codage inverse
- **Code alphanumérique :** ensemble de tous les mots de code utilisé par une machine.

Exemple: ASCII (American Standard Code for Information Interchange)

Systèmes de numération

Le codage

Codes décimaux codés Binaires : DCB

- BCD en anglais est le Binary coded Decimal .
- Chaque chiffre décimal est représenté par un mot de code. Ceci facilite le transcodage pour les E/S.
- Ainsi, à chaque digit décimal est associé un mot de code.
- Pour les 10 digits (0,1,...9) il faut donc 4 bits pour obtenir au moins 10 combinaisons différents $2^3<10<2^4$
- Comme on peut avoir 16 mots de codes différents, 6 mots resteront non utilisés (10->15).
- Un nombre exprimé en DCB occupera une place importante pour son équivalent binaire.

Systèmes de numération

Le codage

Exemple : 792)10

DCB Codage

Base 10	7	9	2
Codage DCB	0111	1001	0010

Systèmes de numération

Le codage

-Table de correspondance :

Décimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

- Application :

Dans les machines à calculer, comme l'entrée saisie et la sortie (affichage) s'effectuant en décimal, le DCB permet d'éviter les transcodages.

DCB→ binaire et binaire → DCB à la sortie.

Systèmes de numération

Code continu

a-Définitions générales :

soit $x=(x_1,x_2,\ldots..,x_n)$ un vecteur binaire de n variables $x_i \epsilon B_2$ et $x \epsilon (B_2)^n$ une valeur x^0 du vecteur x est appelée combinaison ou point de x.

exemple : n=3, $x=(x_1,x_2,x_3)$ → $2^n=2^3=8$ combinaisons de x.

•distance de Hamming :

Soient x^0 et x^1 deux combinaisons de x, on appelle distance de Hamming entre x^0 et x^1 le nombre de bits différents dans x^0 et x^1 .

Systèmes de numération

Code continu

Exemple 1 :

$x^0 = 0100$

$x^1 = 1001$

$d(x^0 , x^1)=d(0100,1001)=3$

Exemple 2 : d(0100,0110)= 1

Systèmes de numération

Code continu

•Combinaisons adjacentes :

Quand $d(x^0,x^1)=1$, les deux combinaisons (ou pts) x^0 et x^1 sont dites adjacentes.

Propriétés : Toute combinaison x* de x (à n dim) est adjacente à n autres combinaisons.

Exemple : n=3 **x=(x1, x2 ,x3)**

Systèmes de numération

Code continu

	X_1	X_2	X_3
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

$d(000,001)= 1 \rightarrow$ 000 et 001 adjacents

$d(x^1,x^2)= 2 \rightarrow x^1,x^2$ non adjacents

Puisque n=3 ➡ toute combinaison est adjacente à 3 autres combinaisons

X^1 adjacente à X^0 ,X^3 ,X^5

Systèmes de numération

Code continu

b- Code continu :

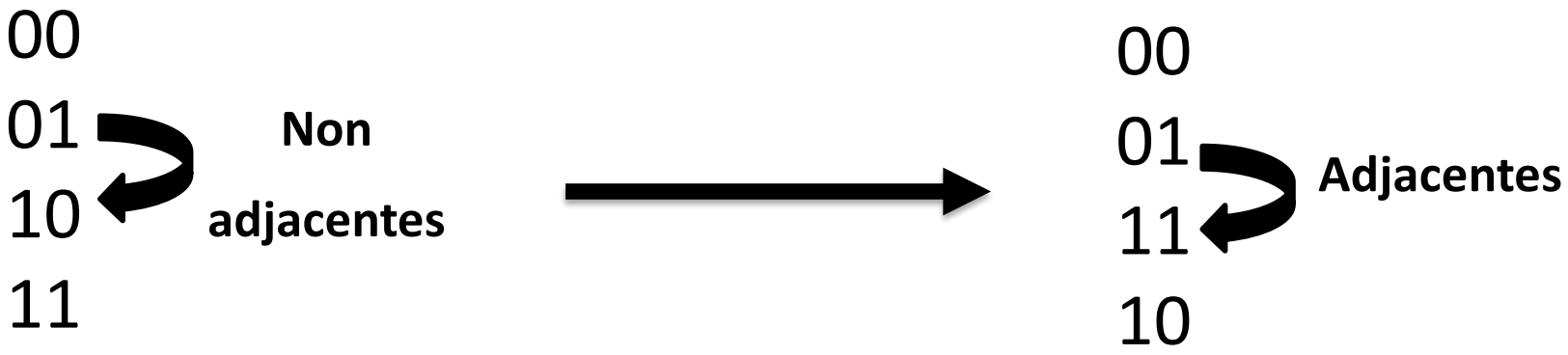
Définition : C'est un code dans lequel deux mots de code (ou deux combinaisons) consécutifs sont adjacents.

Ainsi, on passe d'une combinaison (ou d'un mot de code) à la combinaison (mot) suivante par modification d'un seul bit.

Systèmes de numération

Code continu

Exemple : n=2



Codage binaire naturel

Les chiffres se suivent dans l'ordre naturel (croissant)

Codage binaire réfléchi (continu)

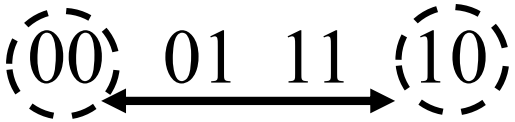
Les chiffres se suivent alternativement dans l'ordre naturel (croissant) et dans l'ordre inverse.

Systèmes de numération

Code continu

C- Code de Gray :

Définition : Code binaire réfléchi cyclique ou le 1er mot est adjacent au dernier.

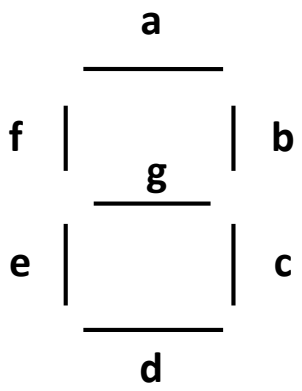


Systèmes de numération

Afficheur 7 segments

Afficheur Numérique à 7 segments

• Ils sont formés de 7 leds. Chaque led à la forme d'un segment groupé dans le même circuit intégré constituant l'afficheur. Chaque segment est relié à une borne et tous sont reliés à une tension commune.



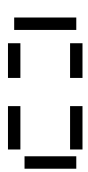
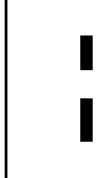
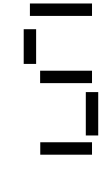
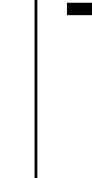
Afficheur Numérique 7segments

Un afficheur numérique permet de visualiser la valeur décimal d'un compteur modulo 10 pour un afficheur 7 segments.

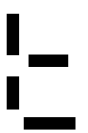
Systèmes de numération

Afficheur 7 segments

Format d'affichage issu des décodeurs TTL :

0	1	2	3	4	5	6	7	8	9
									

Si le décodeur est commandé par un compteur modulo 16 (0 →15) les lettres **'A'** à **'F'** ne sont pas représentés, à leurs places on affiche des symboles :

A	B	C	D	E	F
					**Afficheur éteint**

Systèmes de numération

Afficheur 7 segments

Table de vérité d'un afficheur à anode commune

Anode commune → La tension basse 0 allume le segment. IL existe des afficheurs à cathode commune.

Décimaux	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	1	0	0	1	0	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0

FONCTIONS LOGIQUES ÉLÉMENTAIRES

Fonctions logiques élémentaires

Variables et fonctions logiques

Variables et fonctions logiques : Définitions

- ***Variable logique : U***ne grandeur qui ne peut prendre qu'un nombre fini d'états discrets.
- ***Variable logique binaire ou variable booléenne :*** c'est une variable logique qui ne peut prendre que deux états auxquels on attribue respectivement les symboles 0 et 1 matérialisant toujours des états opposés.

Fonctions logiques élémentaires

Variables et fonctions logiques

Exemples : vrai ‘1’ / Faux ‘0’

Allume ‘1’/ Eteind’0’

Chaud / Froid

5 V / 0V

Ces symboles sont définis par convention et n’ont aucune valeur numérique.

On représente une variable logique par une lettre A, B,…

Exemples : R désigne l’état d’un commutateur normalement ouvert. On décide que : R=1 ⇔ Commutateur ouvert

R=0 ⇔ Commutateur fermé

On travaille généralement en logique positive c-à-d :

Vmax → *code ‘1’*

Vmin → *code’0’*

Fonctions logiques élémentaires

Fonction logique

- Etant données n variables binaires indépendants $x_1,x_2,\ldots x_n$ une fonction logique f des n variables x_i : $f(x_1,\ldots,x_n)$ est une fonction qui, pour chaque combinaison des variables binaires x_i, prend un état binaire bien déterminé '0' ou '1'.
- n variables binaires => 2^n combinaisons binaires possibles et $2^{(2n)}$fonctions de sorties possibles.

Exemples

a	b	c	F(a,b,c)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

F=1 si la majorité des variables sont en 1.

n=3

=> 2^3=8 combinaisons.

Fonctions logiques élémentaires

Fonction incomplètement définie

- Une fonction est incomplètement définie quand sa valeur est indifférente ou non spécifiée pour certaines combinaisons des variables.
  - La valeur de la fonction sera notée : X ou

a	b	c	F(a,b,c)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	X

Rôle :Les variables b et c sont impossibles à réaliser simultanément => simplification des contraintes.

Fonctions logiques élémentaires

- ***Représentation des fonctions logiques***
- La réalisation d'un circuit logique comporte plusieurs étapes dont la représentation du fonctionnement logique.
- Les fonctions logiques peuvent être représentées par :
  - Des tables de vérité
  - Des tables de Karnaugh
  - Des équations logiques
  - Les équivalents décimaux

Fonctions logiques élémentaires

Représentation par les tables de vérité

- Cette représentation consiste à donner la liste de toutes les combinaisons possibles des variabes d'entrées et les valeurs de la fonction.

$$\Rightarrow \text{Table de vérité} = \begin{cases} n+1 \text{ colonnes : n variables+ la fonction} \\ 2^n \text{ lignes: } 2^n \text{ combinaisons} \end{cases}$$

- ***Exemple*** : Fonction inversion = NON

x	$\bar{x} = f(x)$
0	1
1	0

Fonctions logiques élémentaires

Remarque:

Pour n=2 → 3 colonnes ; 2^2=4 lignes.

Pour n=3 → 4 colonnes ; 2^3=8 lignes.

Pour n=4 → 5 colonnes ; 2^4=16 lignes.

.

.

Pour n=6 → 6 colonnes ; 2^6=64 lignes.

Donc la représentation par la table de vérité pour une fonction logique à n entrées (n grand) est pénible.

Fonctions logiques élémentaires

Représentation par les tables de Karnaugh

- Cette représentation consiste à dresser la table de Karnaugh qui se constitue de la manière suivante.
  - On réalise une partition de n variables d'entrées en deux groupes constitués respectivement de p et q variables.
  - si n est pair : p=q=n/2
  - si n est impair : p= (n+1)/2 ; q=(n-1)/2
  - Les 2^p combinaisons sont spécifiées sur des lignes.
  - Les 2^q combinaisons sont spécifiées sur les colonnes.
  - Les combinaisons sont écrites en binaire réfléchi.
  - A l'intérieur de chaque case on porte la valeur de la fonction pour les combinaisons d'entrées correspondantes à la juxtaposition des combinaisons élémentaires associées à la ligne et à la colonne définissant la case.

Fonctions logiques élémentaires

Représentation par les tables de Karnaugh

- ***Exemple 1 : n=3***

Table de vérité

x	y	z	F(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Table de Karnaugh

n=3 : p=2=(n+1)/2 ; q=1=(n-1)/2

Code binaire réfléchi cyclique

z \ xy	00	01	11	10
0	0	0	0	0
1	1	1	1	0